

- -> **loading data from a txt file**
 - -> loading the data from a file into a numpy array, without using Pandas
 - -> in this example, he has a txt file which contains data
 - -> the data is all separated by commas
- -> **np.genfromtxt('data.txt', delimiter=',')**
 - -> this imports the data from the txt file into an array
 - -> this casts it to a float type
 - -> you can set this equal to a variable
- -> **filedata.astype('int32') is another example**
 - -> you can print the file data as an array
 - -> this creates a copy of the data
- -> **filedata = filedata.astype('int32')**
 - -> to load the data from the file
 - -> genfromtxt <- to get data from txt into a Jupyter notebook
- -> **advanced indexing**
 - -> **boolean masking and advanced indexing**
 - -> to learn where in the file data the value is > 50
 - -> for example, you can ask it filedata > 50 <- this returns an entire array full of booleans
 - -> we are asking it to run this boolean operation on every peice of data in the array
 - -> filedata[filedata > 50]
 - -> you only take the data whose values are > 50
 - -> **you can index with a list in numpy**

```
In [179]: filedata = np.genfromtxt('data.txt', delimiter=',')
filedata = filedata.astype('int32')
print(filedata)
```

```
[[ 1 13 21 11 196 75 4 3 34 6 7 8 0 1 2 3 4 5]
 [ 3 42 12 33 766 75 4 55 6 4 3 4 5 6 7 0 11 12]
 [ 1 22 33 11 999 11 2 1 78 0 1 2 9 8 7 1 76 88]]
```

Boolean Masking and Advanced Indexing

```
In [190]: np.any(filedata > 50, axis=0)
```

```
Out[190]: array([False, False, False, False, True, True, False, True, True,
                False, False, False, False, False, False, True, True])
```

- -> np.array([.,.,.,.,.,.])
- -> a[[1,2,8]]
 - -> a is an array
 - -> this returns another array, whose elements are the values at the 1st, 2nd and 8th indices in the a array
- -> **if you want to figure out if those values > 50**
 - -> np.any(filedata > 50, axis = 0)
 - -> this returns an array of booleans
 - -> if you look down the rows, it's telling us if any of the values are 50
 - -> you can also np.app <- there are less Trues in this case

- -> **boolean masking and advanced indexing**
 - -> ((filedata > 50) & f(iledata < 100))
 - -> all of the values between 50 and 100
 - -> it returns out an array which just contains booleans
 - -> whether the elements at that point in the array are True or False
 - -> each element in the returned array is a boolean statement

```
>> a[ 2:4 , 0:2 ]
```

```
>> a[ [0,1,2,3] , [1,2,3,4] ]
```

```
>> a[ [0,4,5] , 3: ]
```

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25
26	27	28	29	30

- -> **indexing question**

- -> we have this matrix, and are being asked to index a certain parts of it
- -> for the second one, you need to use two different lists
- -> we are using indexes to select certain elements in the matrix

- -> **question**

Given a file named data.txt with these contents:

29,97,32,100,45

15,88,5,75,22

What code would produce the following array?

[29. 32. 45. 15. 5. 22.]

```
filedata = np.genfromtxt('data.txt', delimiter=',')
```

```
output = np.any(filedata < 50)
```

```
print(output)
```

```
filedata = np.genfromtxt('data.txt', delimiter=',')
```

```
output = np.all(filedata < 50, axis=1)
```

```
print(output)
```

```
filedata = np.genfromtxt('data.txt', delimiter=',') <- This one
```

```
output = filedata[filedata < 50]
```

```
print(output)
```